Claims: I Claim:

1. A method of generating a proxy web site for wireless devices with micro browser supporting a second markup language, based on a conventional web site created for computer browsers supporting first markup language whereby said proxy web service dynamically interact with the conventional web site for content, the method comprising the following steps:
   (a) interacting with said conventional web site by making a web request in a browser simulator there by capturing the input page of said conventional web site, the destination Uniform Resource Locator (URL) of said web request, the request data being passed to the destination in said web request and the result page of said web request, and marking desired content for conversion within the result page of said web request;
   (b) generating a second input page in said second markup language by identifying a set of user interface elements of the input page involved in said web request, converting said set of user interface elements to equivalent ones in the second markup language and changing the destination URL of the generated input page to a proxy web service;
   (c) generating request object code in the proxy web service to make web request to the destination URL of the web request captured in the simulator passing the input data coming from the web request of the second input page;
   (d) selecting a matching pattern rule within plurality of pattern rules to canonically extract the desired content within the result page of the web request in the simulator and appending the object code for extraction from the selected pattern rule to the proxy web service to extract the desired data from the result of the web request made in the previous step;
   (e) appending result formatting object code in the proxy web service to format the desired data extracted in second markup language using one of the predefined mappings within plurality of mappings.

2. The method of claim1 wherein the step of interacting with the conventional browser in the browser simulator comprises the following steps:
    (a) providing a graphical user interface form comprising atleast three buttons labeled browse, capture and generate, a text box to get URL name and a browser for navigating to the URL ;
    (b) initializing the capture mode to inactive state and waiting for button click events from the browse, capture and generate buttons and invoking the corresponding handler when the button is pressed;
    (c) providing a handler for browse button which will allow the browser to navigate to the URL specified in the text box;
    (d) providing a handler for capture button which activate capture mode and saves the page displayed in the browser as input page;
    (e) providing handler navigate event which is fired when hyperlinks are followed to save request data flowing from the browser in the web request, the destination URL and result page if capture mode is active;
    (f) letting the user highlight the desired content within the result page and saving the highlighted portion in a file.

3. The method of claim1 wherein the step of generating a second input page comprises the following steps:
    (a) identifying the names of parameters passed in the captured web request data in the browser simulator;
    (b) identifying the set of user interface elements within the forms in the page having the same names as the parameters and same destination URL as the captured URL in the web request;
    (c) generating the second input page by converting the input elements to equivalent elements in second markup language and directing the request to a proxy web service.

4. A method of claim 1, where in the step of generating object code in the proxy web service to make web request comprises the following steps:

(a) identifying the hidden and non hidden parameters of the web request in the browser simulator using the input page and the request data captured;

(b) generating the object code for making the web request to the destination URL captured in the browser simulator by substituting the non hidden parameter values coming from the incoming web request into the request data.

5. The method of claim 1 wherein the step of selecting a matching pattern comprises the following steps:

(a) reading the desired content P and the result page R of captured web request in the simulator;

(b) storing the plurality of pattern matching rules which can uniquely identify P within R;

(c) applying pattern rules in sequence to P and R, to find a pattern rule which canonically extract P from R, canonical meaning if the rule is applied to new page R1, the extracted portion P1 is will be having desired content;

(d) adding the object code for extract-pattern procedure of the matched pattern rule to the generated proxy thereby extracting the desired pattern from the result page returned by generated web request.

6. The method of claim 5 wherein the pattern rule and applying the rule to canonically extract the pattern P within the result R comprises the following steps:

(a) finding the pattern P within result page R in the following steps:

1. identifying the hyper text markup language(HTML) element X which immediately encloses the pattern P in the result page R;
2. identifying the type T of the HTML element X;
3. extracting the list L of elements of type T in R;
4. computing the index I of element X in list L;

(b) generating extract-pattern function comprising the following steps:

1. constructing list L of all elements of type T;
2. extracting the element X at index I and returning X to the caller;

(c) allowing the user to navigate to the destination URL captured in the browser simulator with a new set of values for input parameters in the web request;

(d) applying the generated extract-pattern function to the new result page to extract desired data;

(e) allowing the user to verify the extracted result with desired result and returning verification response to the caller.

7. The method of claim 5 wherein the pattern rule and applying the rule to canonically extract the pattern P within the result page R comprises the following steps:

(a) identifying the list of fixed elements in R, the fixed elements being the elements which always appear in the result page independent of the request data passed to the request;

(b) computing the position of beginning and end tags of pattern P relative to a pair of static elements;

(c) generating an extract-pattern function comprising the following steps:

1. locating the position of static elements in R;
2. identifying the position of beginning and end tags relative to the position of static elements;
3. extracting the portion of HTML between begin and end tags;

(d) allowing the user to navigate to the destination URL captured in browser simulator with new set of values for input parameters in the web request;

(e) applying the generated extract-pattern function to the new result page to extract desired data;

(f) allowing the user to verify the extracted result with the desired content and returning the result of verification to the caller if match is found; and

(g) reiteratively repeating steps b to f with a different set of fixed elements until a match is found or all elements are exhausted and returning the result of verification to the caller.

8. The method of claim 1 wherein the second markup language is wireless markup language (WML).

9. The method of claim 1 wherein the second markup language is hand held device markup language (HDML).

10. The method of claim 1 wherein the second markup language is voice XML.

11. The method of claim 1 wherein the second markup language is web clipping (a subset of HTML used in palm devices).

12. The method of claim 1 wherein the second markup language is compact HTML (cHTML).

13. The method of claim1, wherein the generated proxy is a Java servlet.

14. The method of claim1, wherein the generated proxy is an Active Server Page.

15. A software tool for generating a proxy web site for wireless devices with micro browser supporting a second markup language, based on a conventional web site created for computer browsers supporting first markup language whereby said proxy web service dynamically interact with the conventional web site for content, the tool comprising the following elements:

    (a) a browser simulator which will allow a user to interact with said conventional web site by making a web request and marking the desired data to be passed to said wireless devices from the result of said request, the interaction capturing the destination Uniform Resource Locator (URL), the request data being passed and the result page of the web request,

    (b) input conversion means with a translator from first markup language to said second markup language, which can generate second input page in said second markup language with a link to a proxy web service by identifying the user interface elements used in said web request in said simulator to the said destination URL,

(c) request generation means to generate object code in said proxy web service to make a web request passing the data coming from said input page in second markup language,

(d) pattern rule matching means using the browser simulator to select a matching pattern rules within plurality of pattern rules to uniquely identify the desired content within the result page of the web request and add the corresponding pattern extraction code to the proxy web service to extract desired content from the result page of the web request made in the previous step.

(e) result formatting means with said translator from first markup language to second markup language to generate object code for formatting desired data extracted in the previous step to desired format.